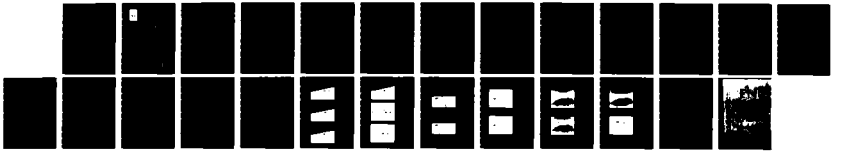
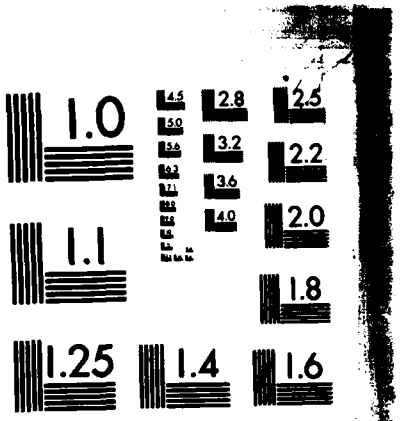
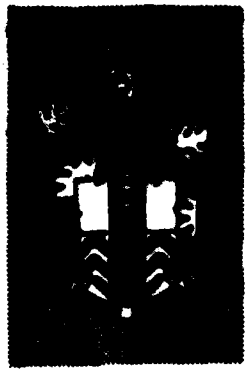
AD-A133 353 NOISE REDUCTION AND SEGMENTATION OF IR IMAGES USING CO-MEDIAN FILTERS(U) ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) S M PEELING JUN 83 1/1

UNCLASSIFIED RSRE-MEMO-3582 DRIC-BR-89250 F/G 17/5 NL





UNLIMITED



RSRE
MEMORANDUM No. 3582

ROYAL SIGNALS & RADAR ESTABLISHMENT

NOISE REDUCTION AND SEGMENTATION OF IR IMAGES USING CO–MEDIAN FILTERS

Author: S M Peeling

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3582

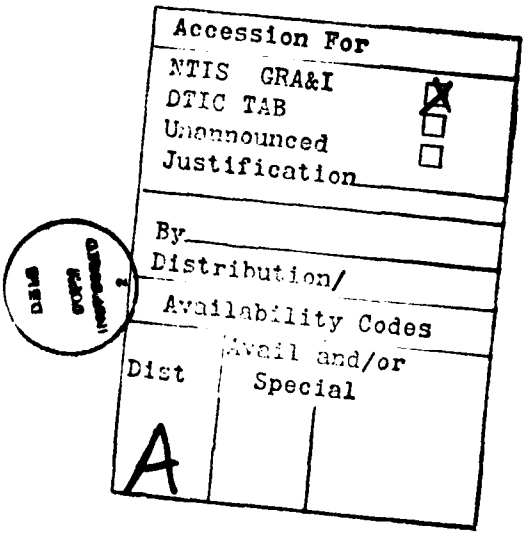
Title: NOISE REDUCTION AND SEGMENTATION OF IR IMAGES USING CO-MEDIAN FILTERS

Author: S M Peeling

Date: June 1983

SUMMARY

A method of separating targets from background in thermal images is described. This includes noise reduction and background estimation prior to target detection. Details of the algorithm developed and its implementation on a microprocessor based image processing system are given. The results obtained from this system are summarised including comparative timings with existing techniques.



RSRE MEMORANDUM NO 3582

NOISE REDUCTION AND SEGMENTATION OF IR IMAGES USING CO-MEDIAN FILTERS

S M Peeling

CONTENTS

1 INTRODUCTION

Increasing sophistication of military hardware is resulting in the need for automatic systems to replace human operators eg in supersonic aircraft where the pilots are required to perform several tasks simultaneously, and in missiles where an automatic system is required to detect and classify potential targets. These systems need to be passive to avoid alerting subjects under surveillance, a requirement which can be met by using infrared sensors which have the capability of day and night operation in most weather conditions (except heavy rainfall). The 10-14 μ IR band is usually used. In IR images exhaust, engines and wheels tend to dominate and the paths of vehicles can be seen.

The algorithms in this paper are specifically concerned with detecting targets in IR images. The segmentation stage of the detection involves using median filters to reject unlikely areas of the image and then selecting high potential regions for targets. The classification stage then involves sizing these areas and rejecting those which are too large or too small.

Before the segmentation stage it is necessary to reduce noise and background (clutter). The noise is assumed to be impulsive (1 or 2 pixels in extent) whereas the background has large scale features (compared with noise and targets) but may have sharp edges.

A standard method for detecting and classifying targets is outlined, in Figure 1A. The image is preprocessed to remove noise, a detection algorithm is applied then the picture is segmented and the target coordinates passed to a classifier.

This paper discusses the more specific scheme of Figure 1B. Here the picture passes through a 2-dimensional 3 x 3 spatial median filter to reduce noise and then a 31 x 31 median filter to provide a background estimator. The outputs from these two are then compared and the result passed to a border following routine which finds the minimum and maximum coordinates of the target which are then passed to a classifier.

2 MEDIAN FILTERS

The median of n numbers, for n odd, is defined as the middle number in size. For n even, the median is defined as the mean of the two middle numbers. In this paper, and in most applications, n is odd. For example, the median of the sequence

37 40 51 41 47

is 41.

Median filtering may be described as allowing a window to move over the picture and replacing the value at the centre of the window with the median of the original picture values within the window. Various forms of filter window can be used eg line segments, squares, crosses etc., some of which are shown in Figure 2. The concept is, of course, readily extendable to percentile values other than 50%.

Points at the edges of the picture can be dealt with in several ways:-

i fewer points could be used to calculate the median near the edge

ii the border points of the output could be left as the, unchanged, values of the input

iii all border points could be set to zero.

The first alternative was not used since the database under consideration had no targets near the edge. The second and third alternatives were used for the 3 x 3 and 31 x 31 median filters, respectively. See sections 3 and 4 for more details.

The implementation of a general nxn, where n is odd, square median filter will now be described, assuming it is operating on an image containing 64 grey levels. The conventional way of calculating the median of these n^2 numbers is to use a histogram ie have a 64 element array and when scanning the window, each time an intensity i occurs increment the contents of the i th array element. (Obviously the sum of the contents of these 64 array elements is n^2). The median is calculated by stepping along the array summing the contents of each location until the sum is greater than, or equal to, $(n^2 + 1)/2$. If this happens at the i th location in the array then the median is set to i. The central pixel in the square, at $((n+1)/2, (n+1)/2)$, is then replaced by the median value. It is necessary to write this value to, say, a separate output line since the original point will be required in the calculation of further medians in the picture.

It can be seen that each median calculation involves n^2 entries to the histogram array and up to 64 additions and comparisons. This is obviously very time consuming. T S Huang et al (4) have proposed a method of speeding up the calculation of medians. This involves calculating the first median as described above and also noting how many histogram elements have values less than the median (ie storing the last and current sums at each stage). Then let

hist = the histogram array

ltmed = the number of values < the median

midpt = $(n^2 - 1)/2$

(note that the midpt used here is one less than that used in the initial calculation). The calculation of the median for the next point along the line can be thought of as moving the whole square one place to the right. Hence the old first column has been replaced, there is a new nth column but all the other points are unaltered. All the intensities in the first column must be removed from hist and each time an intensity is less than the median ltmed must be decremented. The new intensities can now be added to the array hist and each time one of them is less than the median the variable ltmed is incremented. When these operations are complete the new median may be calculated according to which of the two cases apply:

```
1   If      ltmed <= midpt then      (ie median is too small)
    while   ltmed + hist [median] <= midpt     do
            ltmed:  =  ltmed + hist [median]
            median: =  median + 1
    od;
2   If      ltmed > midpt  then      (ie median is too large)
    while ltmed > midpt  do
          median:= median-1
          ltmed:= ltmed-hist [median]
```

For ease of implementation an old point is removed, a new one added and then one of the above formulae applied. Thus with histogram replacement techniques, the time taken in calculating the median is proportional to n (rather than n^2 for recalculation). Hence application of the formulae results in considerable time saving.

As mentioned previously it is not possible to overwrite the central element of the filter square since that element is required in later calculations. In this implementation the images are 256 x 256 pixels and at any time n input lines and one output line are held in arrays. As the window moves across the picture the calculated medians are output to successive positions in the output line. When the end of the line is reached the output line can be transferred to a file. The window now moves down one line so the oldest line in the input array can be replaced by the new line. This saves on storage and also increases the speed of the algorithm.

This section has just listed a few of the more important features of median filtering; a full description can be found in (1).

3 NOISE REDUCTION

The classical approach to noise reduction is to use a linear low pass filter but in certain situations a median filter is better. Two of its main advantages are:

1 Median filtering preserves sharp edges whereas linear low pass filtering blurs them.

2 Median filtering is very efficient for smoothing spiky noise ie noise consisting of impulses narrower than the filter size and with spacing greater than the filter size.

These properties are illustrated in Figure 3 which shows a sequence containing an edge and a negative impulse. After median filtering the edge is preserved and the impulse is removed whereas after moving average filtering the edge has been degraded and the impulse 'flattened'. The pictures in the Appendix also illustrate these differences.

Here, a 3 x 3 square median filter was used since this is large enough to take account of neighbourhood information yet small enough to remove noise spikes. With this filter the edge points are left unchanged since they will be needed at the next stage ie the calculation of the 31 x 31 median. (Setting them to zero would lower the value of the median near the edges).

4 BACKGROUND ESTIMATION AND SEGMENTATION

A median filter is also used as a background estimator and the algorithm described in section 2 is still used. However, the size of the median filter now needs to be such that it is larger than any of the targets in the image. Then the content of the filter window is always predominantly background. For the images used in this project it was found that a 31 x 31 median filter provided a reliable background estimator. A filter size of 15 x 15 was investigated, but for reasons explained in section 6, this turned out to be too small.

The 31 x 31 median is calculated as described previously and is compared with the input pixel (ie the output from the 3 x 3 median). If the difference btween the two exceeds a threshold then the 3 x 3 median is output otherwise $\emptyset$ is output. To save time, the output boundary points are set to zero and not compared (this is equivalent to setting the boundary points to the 3 x 3 median filter values, then comparing and outputting $\emptyset$). Referring to Figure 1B this can be summarised as

$$c = \text{if } (a-b) > \text{threshold then a else } \emptyset \text{ fi}$$

This is described as the segmentation stage. A fixed threshold setting which retained about 1% of the original picture was found satisfactory over the large database described later.

Narendra (6) has shown that a two dimensional median filter can be approximated by two one dimensional filters applied sequentially, with little loss in performance. Since it is currently impossible to implement a 31 x 31 median filter in hardware this work has used a 31 x 1 (ie vertical) median filter

followed by a 1 x 31 (ie horizontal) median filter in one of the implementations. This results in a significant saving in computation time since there are $O(N^2)$ operations in an nxn median filter but only O (2n) in nxl followed by lxn median filters.

At the end of this segmentation stage the aim is to be left with a black background against which potential areas of interest are preserved in grey scale.

5 TARGET ACQUISITION

This stage takes the segmented picture and delivers an nx4 array, where n is the number of targets in the image. The four elements will hold the minimum and maximum coordinates for each target in the order xmin, ymin, xmax, ymax.

The algorithm scans each line of the image looking for non-zero pixels. When such a pixel is found the pixels above and to the left are examined to see if the first pixel is part of a new target or part of an existing target whereupon it will be marked accordingly. All diagonal connections are ignored. The coordinates of the first pixel are used to initialise the corresponding array element, for a new target, or to update, if necessary, the coordinates of an existing target.

With the aid of Figure 4, the algorithm will now be described in more detail. The target count, n, is initialised to zero, the first line of working store is set to zero and the next two lines will hold the first two lines of the image. In general, the working store will contain three consecutive lines of the image but the top line will have already been processed. Referring to the figure, let p be the non-zero pixel under consideration. The first stage is to examine p1 and p2 and select one of the following alternatives:

1 Both p1 and p2 are zero so examine p3 and p4.

   a Both p3 and p4 are zero so p is a single non-zero pixel and as such can be ignored.

   b Either, or both, of p3 and p4 are non-zero. Increment n and replace p, in working store, by the numerical value n (to indicate that it belongs to the nth target). Set the minimum and maximum coordinate values of the nth array element to the coordinates of p.

2 Both p1 and p2 have the same non-zero value or one of them is zero. In either case p is part of the mth existing target (Figure 4b). Therefore p is replaced, in working store, by m and the mth row in the array is updated as necessary.

3 Both p1 and p2 have different (non-zero) values. (Figure 4c). Here p is set to the minimum of 1 and m, say m, and the mth row in the array is reset if necessary after comparison with p and also the 1 th row in the array. The coordinates of 1 are then altered to indicate that it is connected to m by setting xmin to Ø and ymin to m. This situation arises quite frequently when two targets which had, up to this point, appeared to be disjoint, are in fact connected eg the tank barrel in Figure 4d will only be connected to the lower part of the tank body when the point marked with a cross is reached.

It is, of course, necessary in cases (2) and (3) to check that i is not already connected to another element in the array. This is quite easily done by checking whether xmin is zero and if it is by going to the element specified by ymin.

When the picture has been dealt with the array can be condensed to remove all connected targets and also those with either dimension less than 5 or greater than 50. This size criterion removes very small and very large objects which are assumed to be clutter. "Feret" boxes are then drawn round all the valid targets.

6 COMPUTATION SPEED

The image processing system used is shown in Figure 5 and is based on an Intel 8085 microprocessor attached to a 256 x 256 6-bit framestore. The system is I/O driven and the tape recorder allows movement of digitised pictures between the ICL 1906S, mainframe computer, and the system. This is particularly useful when implementing and testing new algorithms since they can be written in Algol 68, tested on the database held on the 1906S and the results output to magnetic tape for display on the monitor of the image processing system. The pictures displayed have 64 grey levels with Ø as black and 63 as white. Both the median filter and contour finding (for target acquisition) procedures were written in Algol 68 and tested on a mainframe. One general median filter procedure was written and used for 3 x 3, 31 x 1 and 1 x 31 median filters.

The heart of the image processing system is an 8-bit Intel 8085 microprocessor system with 16K of erasable ROM for program and 40K of RAM for data. All programs for this system were written in PL8080(5) which is a machine-dependent high level language for the Intel 8080 and 8085 microprocessors. It allows the programmer to retain complete control over the store and register usage but provides the conditional, procedural and loop structure normally associated with a high level language. A PL8080 compiler is available on the 1906S together with PROM programmers (which transfer the compiled program to a PROM) at various locations around the site. Thus PL8080 programs can be written on the 1906S and editted using any of the editors available there.

Procedures exist to transfer data between the peripherals shown in Figure 5.

Implementing the 31 x 31 median in PL8080 was not straightforward because the window contains 961 points with a midpoint of 481 whilst the largest integer that can be held in an 8-bit word is 255. Hence double precision arithmetic is required, for which efficient software is not available. This made the 31 x 31 filter much slower in operation than the 3 x 3, or in fact any filter up to 15 x 15 which would be implemented using single 8-bit words. It was not practical to separate the 31 x 31 filter into two one-dimensional filters since the system only contains one framestore which is required to hold the output from the 3 x 3 median filter for the segmentation stage.

It is worth noting at this stage that neither the median filter nor the contour finding algorithm involve any multiplications or divisions. In fact the latter algorithm involves no arithmetic at all. This obviously helps to minimise the time taken. An appreciable proportion of the time taken by the algorithms involves accessing the frame store. For example, it takes 2 secs to read and replace a 256 x 256 picture in the framestore.

On the 1906S it takes about 67 secs to apply both median filters, the comparison and the border following routine to each picture. This could probably be speeded up since the programs were only really intended to test the algorithms.

On the microprocessor the 3 x 3 median takes about 21 secs, the 31 x 31 and comparison about 3 mins 35 secs and the border following 7 secs to give a total of 4 mins 3 secs per picture.

Now consider the number of operations actually involved in the calculations assuming 50 frames/sec on a 256 x 256 image with 64 grey levels.

1 Medians calculated in the conventional way

a For each pixel with a 3 x 3 median

Address 9 pixels and store in histogram = 18 memory accesses

Average of 32 additions and comparisons = 64 +, comps

b For each pixel with a 31 x 31 median

Address 961 pixels and store in histogram = 1922 memory accesses

Average of 32 additions and compares = 64 +, comps

c Segmentation stage

1 comparison

No of operations = 256 x 256 x 50 x (18+64+1922+64+1)

= 6.8 GOP/s (Giga operations per second)

Replace (b) by 1 x 31 followed by 31 x 1, then for each pixel

Address 15 pixels and store in histogram = 30 memory accesses

Average of 32 additions and compares = 64 +, comps

No of operations = 256 x 256 x 50 x (82+60+128+1)

= 0.9 GOP/s

2 Medians calculated using histogram replacement techniques

a For each pixel with a 3 x 3 median

Address 3 pixels and store in histogram = 6 memory accesses

Average 9 subtractions/additions = 9 +, -

Average 3 comparisons = 3 comps

b For each pixel with a 31 x 31 median

Address 31 pixels and store in histogram = 62 memory accesses

Average 78 additions/subtractions = 78 +, -

Average 31 comparisons = 31 comps

c Segmentation stage

1 comparison

No of operations = 256 x 256 x 50 x (18+171+1)

= 0.6 GOP/s

Replace (b) by 1 x 31 followed by 31 x 1, then for each pixel

Address 1 pixel and store in histogram = 2 memory accesses

Average 3 additions/subtractions = 3 +, -

Average 3 compares = 3 comps

No of operations = 256 x 256 x 50 x (18+8+1)

= 88 MOP/s (Million operations per second)

Hence, assuming that the 31 x 31 median filter is separated into two one-dimensional filters and that histogram replacement techniques are used in the calculation of the medians, devices which are capable of operating at 88 MOP/s are needed.

7 ALGORITHM PERFORMANCE

The algorithms were tested on the Alabama database described fully in (2). This consists of forty three military thermal images containing various combinations of tanks, jeeps and armoured personnel carriers (APCs). The target areas are between 32 and 565 pixels and the contrast ratios ($(I_T - I_{BG})/I_T$) lie in the range 0.12 to 1.00. Each picture consists of 355 x 420 pixels digitised to 8 bits (256 grey levels). As mentioned previously the system shown in Figure 5 uses pictures 256 x 256 pixels with 6 bits. Segments of the original pictures containing the target(s) were used and the intensities scaled down to 6 bits.

For comparison purposes two other methods were tested on the database. They were

1 A simple threshold

2 Linear filters instead of medi filter-

In the first case the mean of the im.. was calculated then each pixel compared with this mean to give Ø if the pixel was less than the mean otherwise the pixel was unaltered. Although this method involves arithmetic it is easy to implement on the microprocessor and takes only 4 seconds per picture.The result shown in Appendix A-3 demonstrates the lack of success in removing the background.

This result is typical of those obtained for other pictures in the database so it will not be discussed further.

In the second case, 3 x 3 and 31 x 31 average filters were used instead of median filters. This was only tested on a mainframe and took about 90 seconds per picture ie half as long again compared to the median filters. Comparisons of the results obtained from median and average filters can be found in the Appendix. Appendix A-1 shows the blurring obtained from a 3 x 3 average filter and A-2 and A-4 demonstrate that average filters are not as successful as median filters at removing background.

Applying both median filters and the contour finding routines to the 43 images, containing 82 targets, in the Alabama database gave 95% detection with approximately one false alarm per picture. Over the same database, the average filters and contour finding succeeded in detecting 96% of the targets, but at the cost of increasing the false alarm rate to 3 per picture. In view of this increase in the false alarms and the time taken by the average filters the median filtering algorithm would seem to give the better results. Also, it is currently impossible to implement a 31 x 31 average filter in hardware. Two one dimensional filters could be used but at the overhead of introducing two divisions.

8 CONCLUSIONS

The system has been tested on several IR databases and also on digitised photographs with encouraging results. The method is equally applicable to conventional tv images where target sizes are smaller than background features, irrespective of grey level histogram.

A 3 x 3 median filter appears to provide a simple yet highly effective method of reducing noise without destroying any of the important information in a picture.

A 31 x 31 median filter provides good background estimation and any difficulties caused by its size can easily be avoided by separating it into two one-dimensional filters with little loss in performance but a marked increase in speed and ease of implementation.

The contour finding algorithm requires a limited amount of working space and no arithmetic. Although the method has previously been used in search-type algorithms, the author believes this is a novel application.

For the type of database considered here one of the main advantages of the routines is the fact that they are rotation invarient. In other applications one might wish to take account of different aspect ratios and there would appear to be no reason why optimum filter sizes could not be chosen for these applications.

It should also be noted that these routines do not require a system with large processing power and hence could actually be used in real systems.

9 FUTURE WORK

A study contract has been placed for the development of an integrated circuit chip to perform sorting operations. In addition to general sorting operations, such as maximum and minimum values, the single IC will be capable of

performing either a 3 x 3 median filter or up to a 31 x 1 median filter in real time. Such chips will allow the construction of small volume systems with low power consumption which will be fast and accurate.

10 ACKNOWLDEGEMENTS

The author has pleasure in thanking Mr J G Harp for his supervision and help during this project, and Dr J B G Roberts for his advice on presentation.

REFERENCES

1 Two dimensional Signal Processing II : Editor T S Huang.

2 Image Processing for target tracking. Second progress report 1 November 1980 to 31 August 1981. J G Harp.

3 Discrimination and Classification of Operating Military Targets in Natural Scenes from Thermal Imagery. Final report of the first project within a cooperative Research program in Image Processing conducted by NATO AC/243 (Panel III) RSG 9. L Sevigny, G Hvedstrup-Jensen, M Bohner, A M Navarro, E Østevold, S Grinaker and J Dehne.

4 T S Huang, G J Yang, G Y Tang, "A Fast Two-Dimensional Median Filtering Algorithm" School of Electrical Engineering, Purdue University (1977).

5 F E Withers, J A McDermid "A Users Guide to PL8080" RSRE Memorandum No 3083.

6 P M Narendra "A Separable Median Filter for Image Noise Smoothing" IEEE Trans Pattern Analysis and Machine Intelligence Vol PAMI-3, No 1, Jan 1981 pp 20-29.

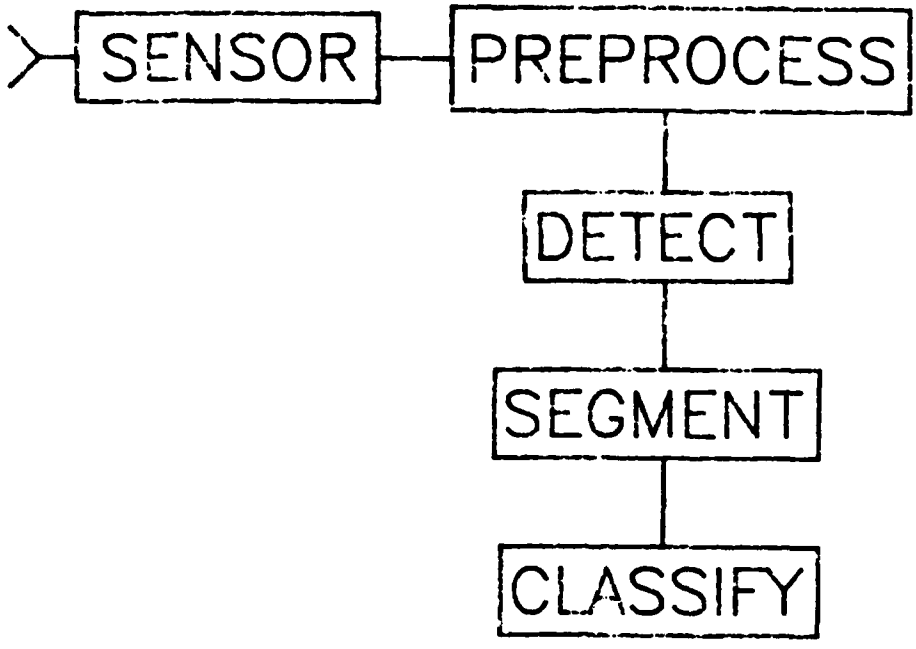


FIGURE 1A

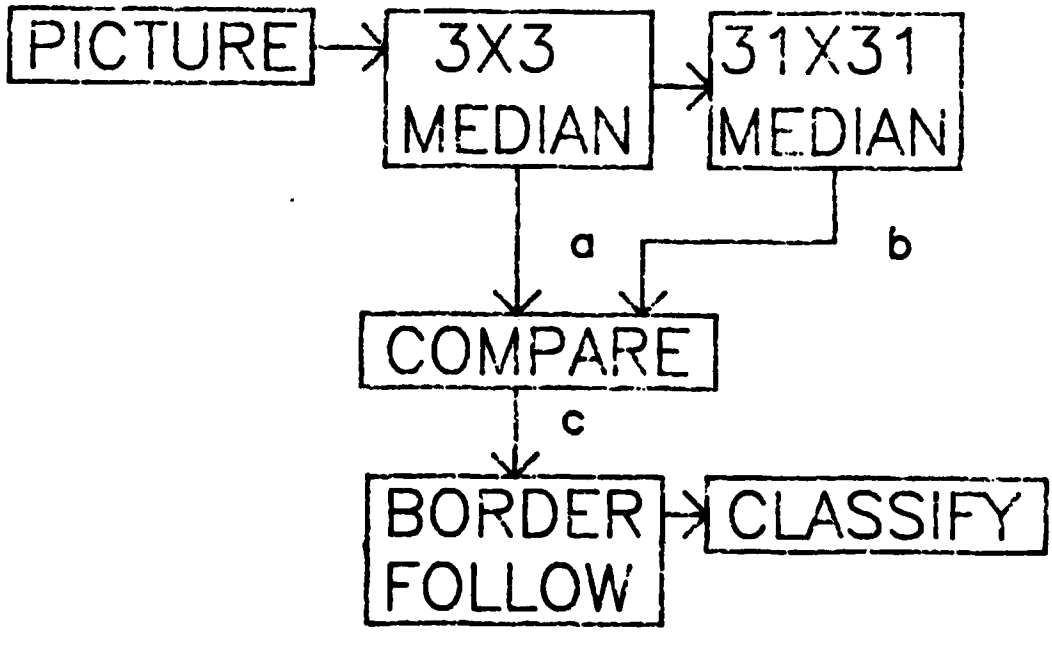


FIGURE 1B

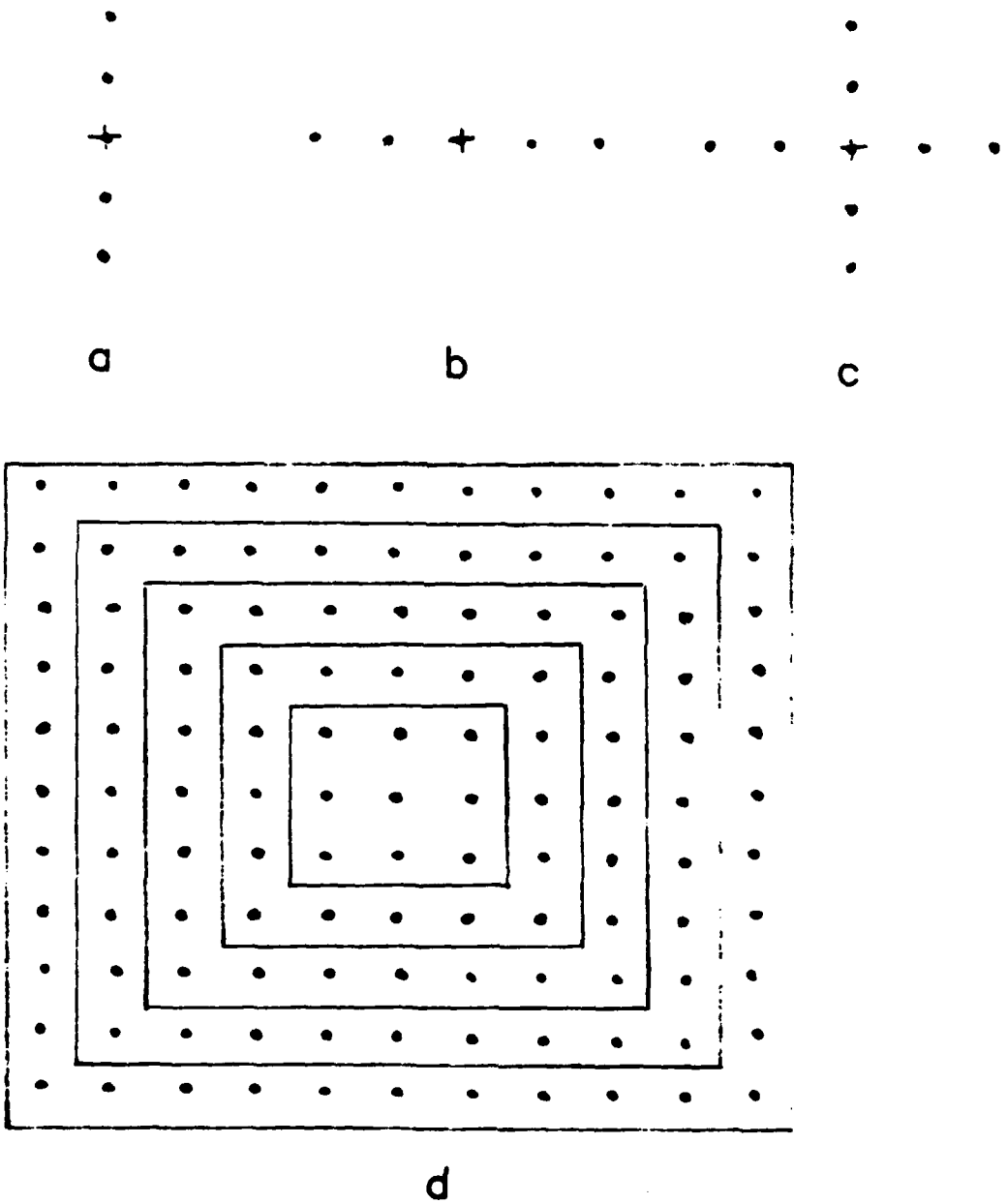


FIGURE 2 EXAMPLES OF MEDIAN FILTER WINDOWS
a,b LINE SEGMENTS
c CROSS
d SQUARES AND SQUARE FRAMES

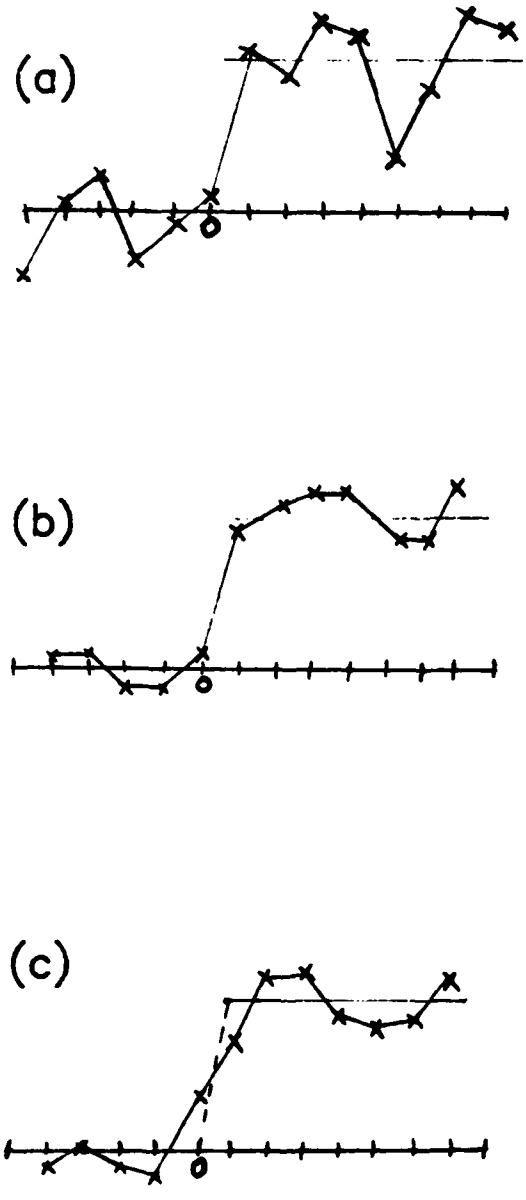
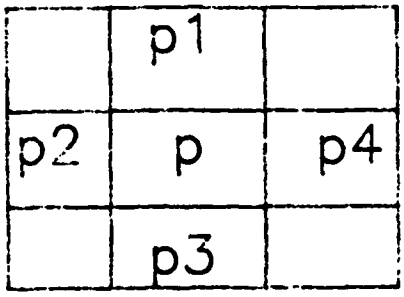
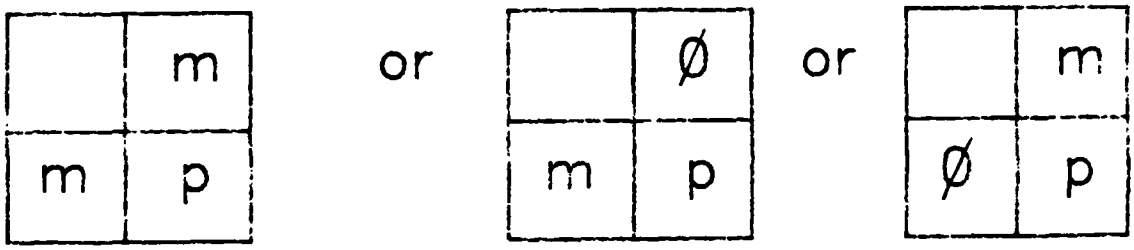


FIGURE 3
COMPARISON OF MEDIAN AND LINEAR LOW PASS FILTERS

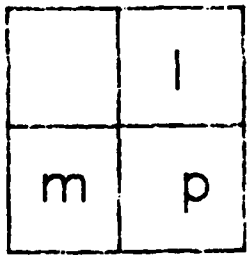
(a) EDGE PLUS NOISE SEQUENCE,
(b) AFTER MEDIAN FILTRERING,
(c) AFTER MOVING AVERAGE FILTERING, N=3



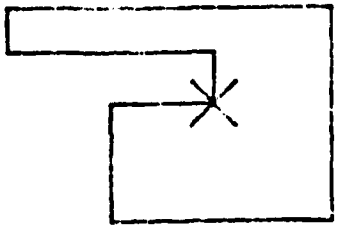
(a)



(b)



(c)



(d)

FIGURE 4

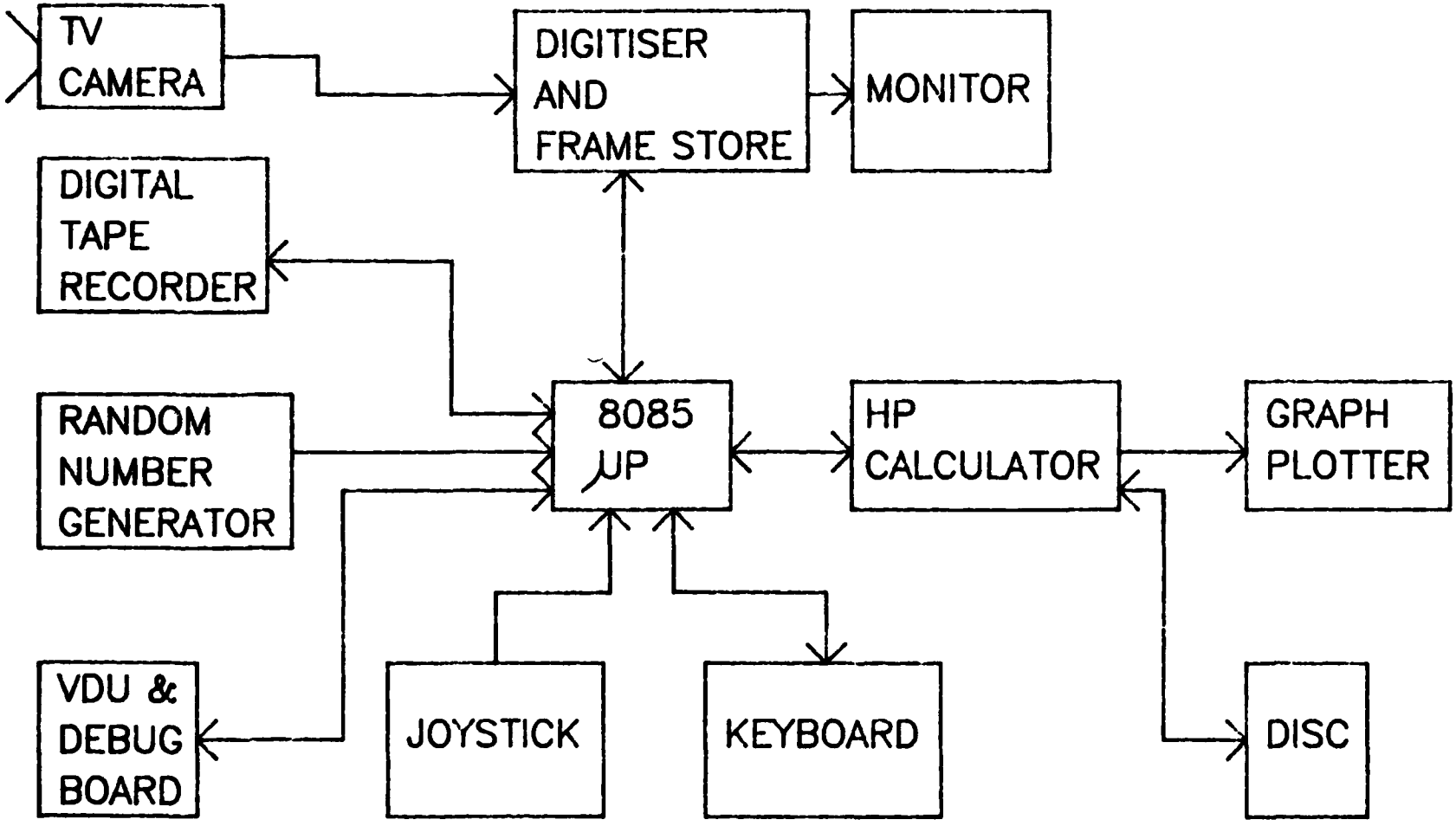
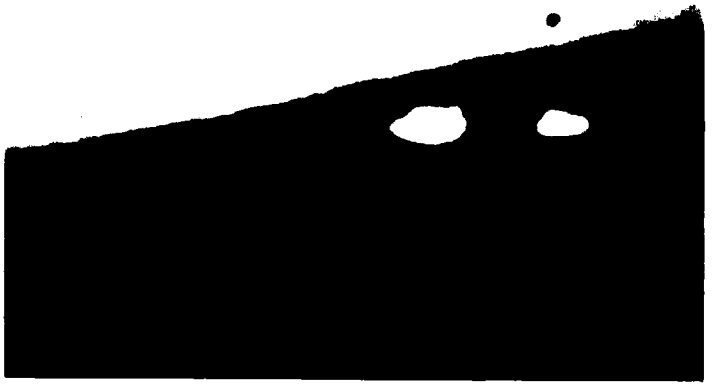


FIGURE 5 IMAGE PROCESSING SYSTEM

ORIGINAL

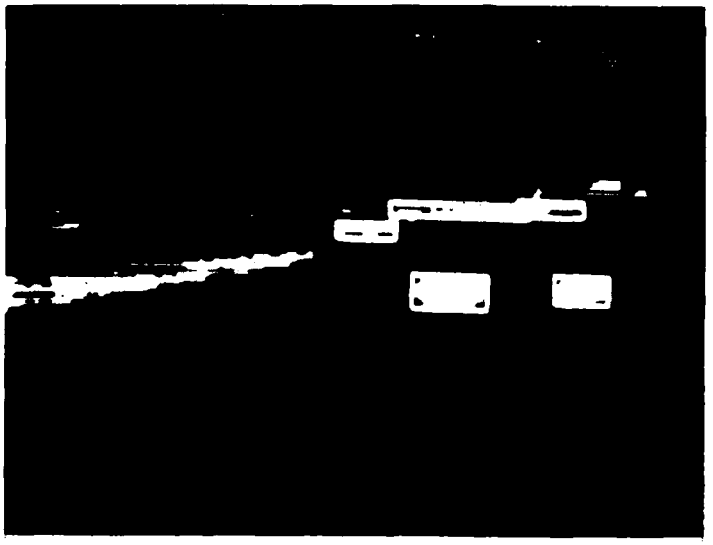


AFTER 3X3
MEDIAN FILTER

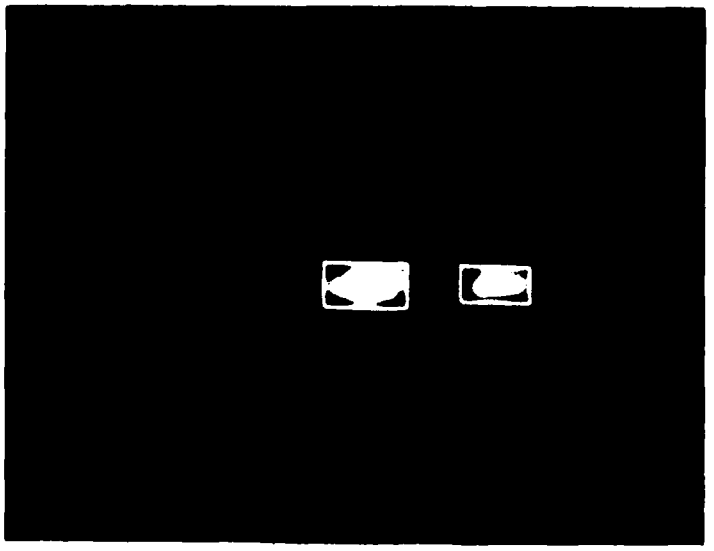
AFTER 3X3
AVERAGE FILTER

ALABAMA 03

ORIGINAL



SEGMENTED AFTER AVERAGE FILTERS (WITH THRESHOLD=14)



SEGMENTED AFTER MEDIAN FILTERS (WITH THRESHOLD=14)

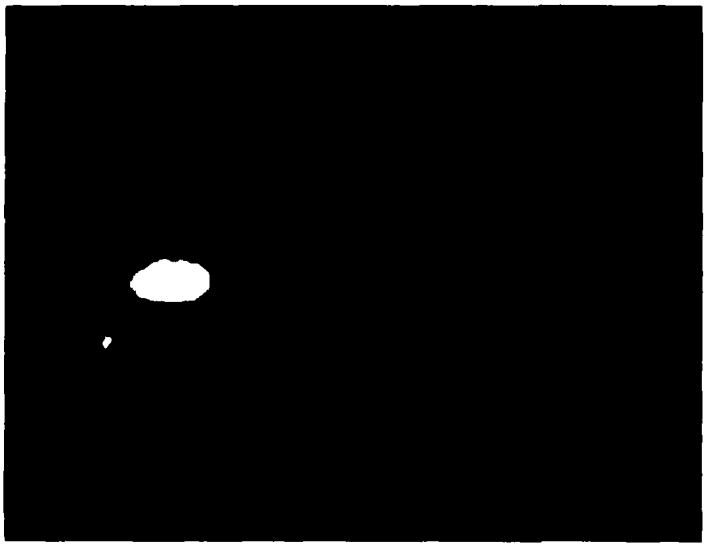
ALABAMA 03

ORIGINAL

THRESHOLDED
WITH MEAN

ALABAMA 01

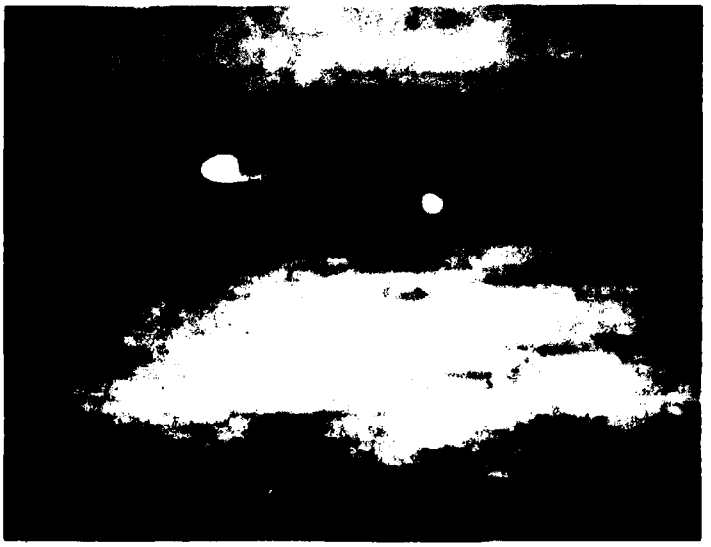
SEGMENTED AFTER AVERAGE FILTERS (WITH THRESHOLD=14)



SEGMENTED AFTER MEDIAN FILTERS (WITH THRESHOLD=14)

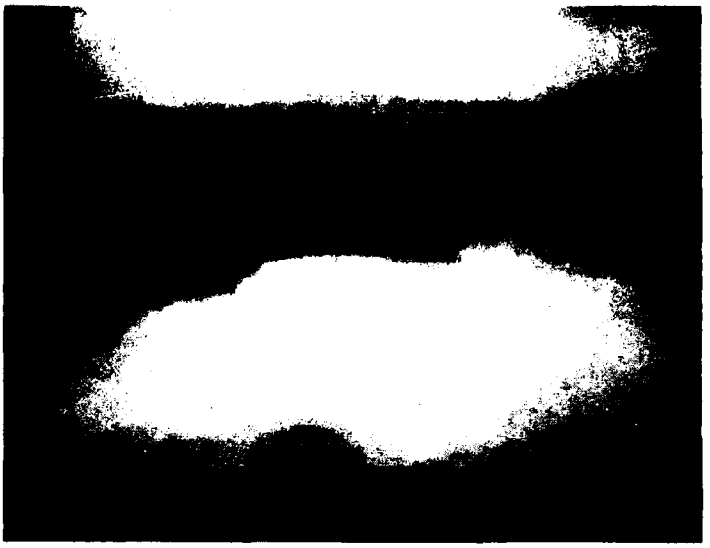
ALABAMA 01

ORIGINAL

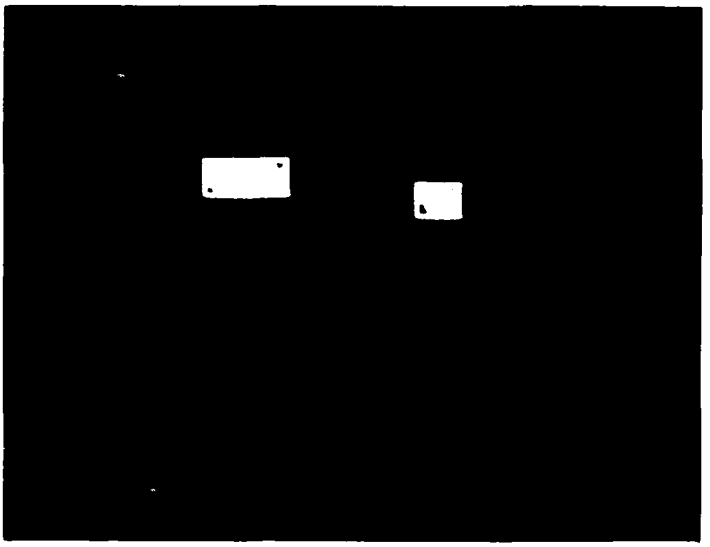


AFTER 3X3
MEDIAN FILTER

ALABAMA 04



AFTER 3X3, 31X1, 1X31 MEDIAN FILTERS



SEGMENTED WITH THRESHOLD=14

ALABAMA 04

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..... UNCLASSIFIED .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification
	MEMORANDUM 3582		UNCLASSIFIED

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location
	ROYAL SIGNALS AND RADAR ESTABLISMENT
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

NOISE REDUCTION AND SEGMENTATION OF IR IMAGES USING CO-MEDIAN FILTERS

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.
PEELING S M				

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract

A method of separating targets from background in thermal images is described. This includes noise reduction and background estimation prior to target detection. Details of the algorithm developed and its implementation on a microprocessor based image processing system are given. The results obtained from this system are summarised including comparative timings with existing techniques.


FIL